Specifications for Upgraded AA (AA+) based Systems

Editor: R. Raible
Contributors: G. Robbins
J. Redfield

PRELIMINARY

03/17/92	first cut at it
03/20/92	added pin descriptions/loading specs.
03/27/92	fleshed out features using grr document of 3/26
04/08/92	added grr's features
04/09/92	added grr's new regbits
04/10/92	final cleanup before release to team
04/20/92	added Hedley correction & new header
05/04/92	added DMA slot charts
05/05/92	added COPWAITH, AGAIN pin, individual sprite DMA enables
05/07/92	swapped optional XMIT FIFO for optional UFHRES serial
05/08/92	final changes, including TVL's inputs, renamed APE & LIKE
05/11/92	more final changes: renamed STUPID,redefined SLOTCTL & write-posting
05/12/92	yet more: conformed SLOTCTL to grr's expectations
05/14/92	added BPLCON5 containing extra horizontal scroll bits

Table of Contents

1. Summary of New features for AA+

The AA+ chip set consists of 3 chips which replace many components found in AA as illustrated in the table below:

AA	AA+
8374 - ALICE	ARIEL - 160 pin, 1u CMOS custom
8364 - PAULA	
4203 - LISA	BELLE - 84 pin, 1u CMOS custom
VIDEO DAC	
8520 - VIA (2)	DEBI - 160 pin, 1u CMOS gate array
GAYLE	
sundry TTL	

The first custom chip, referred to as ARIEL (Agnus/Paula Element), implements the timing, control, address generation, peripheral interface, and audio output functions previously implemented primarily by the existing AGNUS (ALICE) and PAULA chips. It is implemented as a full-custom CMOS design in a 160 pin package. The design could be characterized as a re-implementation of the merged ALICE and PAULA chips in 1u CMOS with major enhancements.

The second custom chip, referred to as BELLE, implements the video display functions currently implemented in the LISA chip with an on-board triple 8 bit video DAC. It is implemented as full-custom CMOS design, in a 84 pin package. For applications requiring the full 24 bit RGB digital video bus, the DAC would be readily excised from the data base and the digital video bus would be brought out to pads (requiring a 144 pin package).

The system control ASIC, which is expected to vary somewhat depending on the target system, incorporates processor interface and system control logic currently implemented in the GARY / GAYLE chip families and also the peripheral ports and counter/timer functions which currently reside in two 8520 VIA port chips in the Amiga system implementations.

The DEBI chip would be implemented as a CMOS gate array in a 160 pin package. It would be a relatively low gate count design with moderate speed requirements. The processor interface and system control logic is not radically different from that implemented in the A600 GAYLE chip, but the 8520 VIA designs would need to be ported and re-implemented in the gate array technology.

Below is a summary of the features new to this chip set:

- Doubling the AA chip memory bandwidth (8 times original Amiga Chip Set) enhances processor and coprocessor access to chip RAM for all supported display modes.

- 57 Mhz clock input allows for clean definition of EXTRAHIRES (EHRES) pixels. BELLE's and DEBI's clock generators are synchronized by

means of ARIEL's A_SINC output. The genlock CLK_EXT and CLK_EXT_EN_N pins have been implemented on the BELLE chip. Since we do not desire to modify the current genlock standard, a means must be provided to double the genlock clock output from 28.6 MHz to 57.2 MHz, either on the BELLE chip or on the PCB.

- The basic Amiga chip bus cycle has been shortened from 280nS to 140nS, only multiple CAS cycles remain at 280nS.

- Triple 8 bit video DAC has been integrated into the BELLE chip.

- 1.76M Floppy Drive capability has been added.

- The serial port has been modified so that it is capable of receiving baud rates in excess of Midi rates (4 byte FIFO).

- A CRC accumulation register allows relatively painless calculation of CRCs during floppy accesses.

- Both chip RAM and register address space has been expanded.

- SLOTCTL has been added, allowing the quantity of fixed DMA slots per line to be tailored to the horizontal scan rate. This allows increased overscan and/or viewport scroll capabilities.

- The CPU can access all ARIEL registers without chip databus contention (optional)

- ARIEL supports write-posting of chip memory and chip registers (optional) (would exclude INTENA register so as to avoid spurious interrupts)

- Blitter operation has been enhanced (blitter B source masking and next to last word mask) (optional).

- Serial port XMIT capabilities have been enhanced (4 byte FIFO) (optional).

- Hooks have been added for future external high bandwidth audio (optional).

- Individual sprite DMA enables have been added (as required by graphics).

- Hooks have been added for future external video decompression via A_AGAIN_N pin on ARIEL (optional).

- A register has been added to ARIEL (COPWAITH) containing additional

lines greater than 255.

2. Description of New Features

2.0 Intro

This section is intended to serve as a description/definition of the changes to the Amiga chip set from the existing AA version to an enhanced version known as AA+. This is an architectural document - it describes the essential features from a chip set perspective, including those that are optional and whose implementation will be at the discretion of the VLSI designers responsible. Defining these features at this point insures both that the fundamental implementation issues are reasonably well understood and also provides an opportunity to implement the supporting details in the first pass so that future enhancements can avoid the obstacle of requiring lock step changes to the entire chip set. The above-mentioned optional features will be set in italics to help differentiate them from the rest of the feature list.

2.1 AA+ Chip Set Design Goals

The stated goals of the AA+ chip set effort are to provide a competitive product in terms of video capabilities (defined by effective VGA/SVGA and 72 Hz operation), floppy disk capacity/compatibility (PC 1.44 MByte format) and conversion to a (hopefully) reliable CMOS design. The new chip set is to be implemented in a reasonably short time frame. This requires care in the selection of which features are to be implemented and expediency in their definition and means of implementation. However, this should not be to the exclusion of concern for extensibility and the goal of creating functional blocks that can be re-used in the future.

2.2 Descriptions of New and Revised Features

2.2.1 Global Performance Improvements

There are a number of items best addressed in terms of their global performance impact than the particular functional areas changed. These include changes to the basic technology, modifications to the basic Amiga cycle time or organization and extension of enhanced fetch modes (32 bit, double, quad-CAS) to other DMA sources.

Shorten Basic Amiga Chip Set Cycle Time

The basic Amiga chip bus clock period is changed from 280nS to 140nS (NTSC). This constitutes the simplest way to take advantage of the faster DRAM cycle times and basically doubles processor, Blitter and Copper performance in traditional modes, and during the single-CAS cycles of enhanced modes. The AA+ double and quad CAS cycles take two contiguous cycles, of which only the first has an RGA code. Note that the double-CAS

cycle could be handled as two single CAS cycles, but there's
non-linear relation between single, double and quad-CAS operation and constraints.

This feature has two major effects on system performance. First it improves throughput by effectively doubling the number of cycles available in standard low bandwidth display modes and the retrace periods of the new high-bandwidth display modes. This extends the "transparent" processor / Blitter access that characterized the original Amiga low-resolution display modes to higher resolutions. Also, since processor cycles can now be scheduled at 280 nS intervals, it provides a memory cycle that can support a 14MHZ processor running full speed from chip memory, at least as long as cycles are available.

A problem with global performance improvements is that they raise the specter of incompatibilities with existing software, mostly poorly written games, which rely on the existing performance characteristics of the Amiga chip set and processor interface to throttle play rate or schedule activities. To a degree, such software can be considered already broken, since it may fail in A500 / A590, A600 / PCMCIA, A2000 , A3000 or A1000+ systems where fast memory, 32 bit chip memory and faster processors provide a higher performance level than the A500 / A600 baseline. However, since the immediate market acceptance of a new machine depends on a perception of software compatibility and availability (especially at the low end), it is still desirable to address the issue to some degree. The position that we take is to provide register compatibility with the existing Amiga chip set and provide control bits that effectively disable the enhanced performance features to provide near-equivalent performance with the A500 / A600. The benefits of this are fairly obvious, but is worth mentioning that anything that improves processor access to chip memory and/or Blitter performance is extremely important, since these are areas in which the new video modes not only impact system performance, but also require additional performance to support the data requirements of higher resolution video modes effectively.

2.2.2 Processor Interface

Control Signal Changes

The old separate _RAMEN/_REGEN signals have been replaced by a single chip select line (A_CS_N) and a RAM versus REGISTER selection pin. This saves a pin on DEBI since _RAMEN and _REGEN had the same timing requirements and the selection can be just a processor address line. The _DBR/_BLIT (DRAM bus required) has been redefined by a A_DONE_N pin. The old signal merely indicated that the Amiga chips didn't need the chip bus on a particular cycle. While this used to be an adequate "done" indication, it isn't adequate in the light of the write posting feature described below. With respect to the AA ALICE chip, the _AS, _UDS and _LDS pins are back (P_AS_N,P_UDS_N, AND P_LDS_N respectively). These are needed if ARIEL is to directly support the processor bus interface and handle direct _RAS/_CAS generation with byte write selection.

Certain pins have alternate definitions depending on whether a 16 bit (68000) or 32-bit (68020/030) bus interface is selected. The 68000 defines cycles in terms of _AS, R_W, _UDS, and _LDS, while the 68020(030) uses _AS, R_W, A1, A0, SIZE0, SIZE1, and _DS. Since the 68020(030) uses more pins, grounding the 68030 _DS pin during reset selects the alternate 68000 pin definitions:

68000	68020(030)	AA+ Name / Description	
_AS	_AS	P_AS_N	- same (address strobe)
-	_DS	P_16_BIT_N	- address strobe or mode select
_UDS	_SIZE1	P_UDS_N	- upper data strobe / size code
_LDS	_SIZE0	P_LDS_N	- lower data strobe / size code
_DR_32BIT	A0	DR_32BIT_N	- chip bus width (16/32) / byte select
A1	A1	P_A(1)	- same (word select)

The _BLS pin is deleted since it is really only reflects the presence of any outstanding processor cycle request and can be internally derived. An P_QUAD_N (pin not needed since there are no plans to implement feature) pin is added to request processor side multi-CAS cycles nominally for DMA, but perhaps also usable for '030 burst mode accesses with cycle matching buffers. To properly support this, the addressincrementing on multi-CAS cycles should actually increment with wrap the LSB's of the supplied address, not just OR in the increment as it does now for double-CAS cycles.

Data Bus Interface

The ARIEL chip directly supports coupling of the 32 bit chip data bus to a 16 bit processor bus and provides control signals for an external 16 bit latching buffer to support 32 bit processors.

Write Posting (optional)

To minimize system stalls caused by chip access, ARIEL supports write posting (aka write buffering). What this means is that for a chip register or memory write ARIEL latches the processor address and data and immediately asserts A_DONE_N, allowing the processor to proceed while the write is completed at some later time. Obviously, if a read or a second write request is issued before the first one has been executed, the second request must be deferred until completion..

There is some concern about potential adverse software impact, but this is probably minimal since the write will be completed just as soon as if the processor was stalled on it, and any subsequent read (or write) will cause resynchronization by stalling on an uncompleted write. However, it is important that writes to INTENA bypass this feature since this would leave open the possibility that interrupts would be recognized after they were disabled by the CPU. Write posting requires no significant on-chip resources, but entails some elaboration of the processor interface logic.

Fast Register Access (optional)

Since the ARIEL chip integrates the 16-bit bus gateway, it is possible to provide a path to the Amiga chip registers (at least those in ARIEL) that does not require arbitration for the external chip data bus and the resultant contention with video fetches and other chip cycles. Since a certain percentage of operations relating to interrupt control and serial control require access to chip registers, even when running out of ROM or fast memory,

access requires in effect dual-porting of the on-chip registers either via a dual-bus approach or time-multiplexing the existing data paths.

Address Space Expansion

The Address Bus path and internal registers are extended to support at least 8 MByte of chip RAM. It might make sense to provide an even larger space when implementing these functional areas, however it's not obvious supporting that more than 8M-byte chip memory will be economically reasonable.

2.2.3 Blitter Enhancements

The Blitter performance has lagged even further behind than the processor interface in supporting the increased data handling requirements of the new video modes. The mechanism designed to support NTSC images in modes that leave 50% of the memory bandwidth available for Blitter use does not work effectively on high resolution 8 bitplane images where only retrace interval cycles are available for processing. Short of redesigning the Blitter to take advantage of the extended fetch modes, there are two approaches that help to mitigate this problem: one is to increase the number of cycles available for the Blitter, the other is to modify the Blitter in relatively small ways to help with certain known inefficiencies.

Blitter B Source Masking (optional)

The current operand mask set (first word, last word masks) can only be applied to the A source. For certain classes of blits it is desirable to only use the B source, and having to use the A source just for masking requires additional cycles. The change that the software people have requested is to provide a control bit that indicates that the masks should be applied to the B source instead of the A source.

Imagine that we wish to fill an arbitrarily bit-aligned rectangle with 1's, without affecting any surrounding bits. The way this is done now is to set the BLTADAT register to $ffff, and the BLTAFWM and BLTALWM registers to mask the source. D is set to the destination, and B is set to pre-read the destination. The only DMA sources enabled are B and D. Unfortunately, the fastest mode of operation of the blitter is with A and D. With A and D enabled each word blitted takes 4 ticks. With D and B enabled it takes 6. With B masking we could use the faster D=A blit for fills and scrolls.

Next to Last Word Mask (optional)

An additional mask word is needed to efficiently perform some of the frequently used blit operations involving operand shifts. Lacking this mask, multiple Blitter passes are needed to complete the logical operation. An additional register is added to provide this next-to-last work mask value, also a control bit is needed enable the mask (for compatibility).

The next to last word mask would more than TRIPLE the speed of masked (cookie-cutter) blits. These blits are the most important ones for the typical game. Note that it is desirable that this mask could be applied to the B source as outlined above for the first and last word masks.

2.2.4 Copper Enhancements

Extended RGA Space

The RGA address field in the Copper move instruction is extended to support up to 10 bits of Copper address space (1024 word registers - this is required if more than one or two registers are added by the proposed changes).

Improved Copper Wait Instruction

A COPWAITH register has been added which supplies high order vertical position and enable bits for the Copper WAIT instruction. The bits are reset every vertical blanking interval so as to transparently support old copper lists.

2.2.5 Floppy Enhancements

Floppy DMA Bandwidth Enhancement

The floppy interface is enhanced to take advantage of extended fetch and/or 32 bit cycles to support the throughput requirements of high density 1.76 MByte floppy drives.

1.76 MByte Data Rate

The floppy PLL and related circuitry is enhanced to run at 2 times the current speed to support the faster bit rate associated with high density drives. Four registers are provided to specify the PLL magic constants used to optimize loop performance in the high-density mode.

CRC Generator/Checker

A programmed IO, register based CRC accumulation device is implemented in the system control gate array. This has some performance impact versus a direct hardware implementation, but does allows the processor to implement the decisions on which bytes are included in the CRC for a given format.

2.2.6 Serial Enhancements

Serial FIFO Extension

The serial port has 3 stages of FIFO buffering added to improve performance in the long interrupt latency of the Amiga system environment. This results in a total of 4 stages.

Serial Port Xmit FIFO (optional)

The serial port has 3 stages of FIFO buffering added for a total of 4 stages.

2.2.7 Audio Enhancement

Internal Audio Disable (optional)

The audio DMA fetch mechanism is enhanced to take advantage of extended fetch and/or 32 bit cycles to support the throughput requirements of 16 bit audio samples at 44.1 KHz sample rates. A register bit disables the on-board DAC output, allowing a future external DSP / DAC device to sit on the chip bus and provide compatible CD quality audio.

2.2.8 Video Enhancements

The video features described address two basic areas: supporting the scan rates and screen formats requested by management, and providing what we feel are competitive features for the Amiga to be considered as a state of the art multi-media, game, and entertainment system.

Doubled Pixel Clock

The video data path is enhanced to support a 57.2 MHz pixel clock which allows up to 800 by 600 SVGA displays at 72 Hz non-interlaced refresh rates.

Video DMA Bandwidth Enhancement

The video buffer/shifter is enhanced to support quad-CAS cycles, doubling the available bandwidth versus the AA chip set or 8 times the original Amiga bandwidth. The quad-CAS cycles can also be used to improve the efficiency of lower resolution modes where desired.

Video Decompression (optional)

The A_AGAIN_N pin on ARIEL allows external logic to control whether or not the bitplane pointers are incremented after each fetch. If the pin is low then the same bitplane data is fetched again.

Individual Sprite Enables

When using the sprite foldback DMA slot allocation, it may be possible that sprite fetches would utilize all of the free DMA slots, thereby leaving none for the copper. This would be catastrophic in that one would not be able to set up the next viewport. A means must be provided to limit the number of sprite DMA time slots so that this can be prevented.

DMA Timeslot Reallocation

The current DMA scheduling mechanism involves the assignment of fixed DMA timeslots at the beginning of each scan line. In the display modes that require fast scan rates all of the slots are not needed on each scan line and the fixed overhead of these slots defines the minimum retrace period and the amount of scrolling possible. The slots for DRAM refresh, floppy disk and audio DMA are fixed and cannot be preempted. The slots for sprite fetches follow and can be sacrificed for additional display width. DMA slots alternate with free slots for copper/blitter access. The DMA timeslot reallocation feature provides an option of replacing the current fixed slot

allocation with a more compact one. As an additional option, sprites can be folded back into the odd time slots at the beginning of the scan line. This allows bitplane fetches to start even earlier, immediately after the audio fetches. Note that, as before, Copper and Blitter do not have priority and it is possible for them to be locked out during display lines.

SYSCTL **XXX** W AB System Control Register

This register contains various bits that control some of the AA+ performance enhancements for compatibility's sake and some other random control bits. All bits are set to zero by reset.

Bit#	Name	Function
15	**GOFAST**	Effectively enables 140 ns cycles
14	**FASTREG**	Enables fast register accesses (optional)
13	**WPOSTMEM**	Enables memory write posting (optional)
12	**WPOSTREG**	Enables register write posting (optional)

note: when GOFAST is not set, the cycle arbitration acts as if cycles can only start on multi-CAS boundaries, thus effectively disabling the enhancement due to the 140 nS cycle times.

BLTCON1 042 W A Blitter Control Register 1
(these are extra bits for the existing BLTCON1)

Bit#	Name	Function
11	**NLWME**	Next to Last Word Mask enable (optional)
10	**BMASKE**	B source masking enable (optional)

BLTANLWM 076 W A Blitter Next to Last Word Mask (optional)

(combined description with BLTAFWM/FLTALWM)

The patterns in these registers are "anded" with the first, next to last and last words of each line of data from Source A into the blitter. A zero in any bit overrides data from Source A. These registers should be set to all "ones" for fill mode or for line drawing mode. For the sake of compatibility, the Next to Last Word Mask register is treated as all "ones" unless the enable bit in BPLCON1 is set.

On narrow blits where first, next to last, or last words overlap, the mask values specified for overlapping words should be identical for consistent results. When the BMASKE bit is a "one", the masks are applied to the B Source instead of the A Source.

COPINS 08C W A Coprocessor Instruction Fetch Identify

(add to table)

BIT#	IR1
15	-
14	-
13	-
12	-
11	**DA11**
10	**DA10**
9	**DA9**

FMODE 1FC W AB Video Memory Fetch Modes

(add to table)

Bit#	Name	Description
09	**SPQUAD**	Sprite Quad-CAS mode

ADKCON2 XXX W A Audio, Disk, UART Controls 2 Write
ADKCON2R XXX W A Audio, Disk, UART Controls 2 Read

These register contain control bits for the Audio, Disk and UART functions.

Bit#	Name	Description	
15	**SET/CLR**	...	(optional)
13	**PPLLE**	Floppy Disk Programmable PLL Constant Enable	
12	**MEGA**	Floppy Disk Clock Rate Control (see table)	
9	**DSKPAGE**	Floppy Disk Page mode (Double CAS)	
8	**DSK32**	Floppy Disk 32-bit mode	

DSKPAGE	DSK32	Fetch Incr.	Bus Width
0	0	1	16
0	1	2	32
1	0	2	16
1	1	4	32

FAST	MEGA	Clock	Use
0	0	4 uS	GCR
1	0	2 uS	MFM
0	1	1 uS	MFM (2 MByte)
1	1	-	-

DSKPMIN XXX W A Floppy Disk PLL Phase Minimum Value
DSKPMAX XXX W A Floppy Disk PLL Phase Maximum Value
DSKPFIN XXX W A Floppy Disk PLL Frequency Minimum Value
DSKFMAX XXX W A Floppy Disk PLL Frequency Maximum Value

When enabled by the PPLLE bit in ADKCON2, these register provide programmable tuning values for the floppy disk PLL constants. This is intended to optimize the loop characteristics for 2 MByte MFM, which may or may not need changes to the existing fixed constants.

FMODEA XXX W A Audio Channel Fetch Modes (optional)

This register contains the fetch mode bits for the audio channels.

Bit#	Name	Definition
15	**AUD3RWM**	Audio Channel 3 Read/Write Mode (capture)
14	**AUD3QUAD**	Audio Channel 3 Quad-CAS Mode
13	**AUD3PAGE**	Audio Channel 3 Page Mode
12	**AUD3F32**	Audio Channel 3 32-bit Mode
11	**AUD2RWM**	Audio Channel 2 Read/Write Mode (capture)
10	**AUD2QUAD**	Audio Channel 2 Quad-CAS Mode
9	**AUD2PAGE**	Audio Channel 2 Page Mode
8	**AUD2F32**	Audio Channel 2 32-bit Mode
7	**AUD1RWM**	Audio Channel 1 Read/Write Mode (capture)
6	**AUD1QUAD**	Audio Channel 1 Quad-CAS Mode
5	**AUD1PAGE**	Audio Channel 1 Page Mode
4	**AUD1F32**	Audio Channel 1 32-bit Mode
3	**AUD0RWM**	Audio Channel 0 Read/Write Mode (capture)
2	**AUD0QUAD**	Audio Channel 0 Quad-CAS Mode
1	**AUD0PAGE**	Audio Channel 0 Page Mode
0	**AUD0F32**	Audio Channel 0 32-bit Mode

SLOTCTL XXX W A Scan Line Fixed DMA Format

This register defines the ordering of the fixed DMA slots at the beginning of each horizontal scan line. Formerly, fixed DMA slots occurred on consecutive even cycles of the horizontal counter, and odd cycles were not utilized. This is illustrated in the first column of the table below. Now some measure of control can be exerted over the position and quantity of the various DMA categories. The SPFOLD bit allows the sprites to start earlier using the initial odd cycles. NOLOLS signifies that the STRLONG strobe is not needed so that floppy DMA can preempt the second refresh slot. The remainder of the bits define the quantity of each DMA category. This allows fine tuning DMA requirements, to the point of allowing the floppy, DRAM refresh, and the audio channels to get fractional slots on a line-averaged basis. Note that AUDSLx determines the TOTAL number of audio slots per line, and that not all channels may be serviced on any given line, but as in the case of SVGA below, alternate on a 2 line or 4 line rotation.

Bit#	Name	Definition
14	**RFSL2**	This 3 bit field defines the total number of DMA slots reserved
13	**RFSL1**	for both refresh and floppy disk. All slots not required by refresh
12	**RFSL0**	are available for floppy DMA. Default is 7 (binary 111)
11	**LOLREQ**	Guarantee presence of second refresh slot for LOL strobe
10	**SPFOLD**	Enable the sprite foldback mode (see text above)
09	**AUDSL1**	This 2 bit field defines the number of audio DMA slots per line,
08	**AUDSL0**	according to the following table:

AUDSL	Number of Slots
00	4 (default)
01	1
10	2
11	0

07-03	-	unused
02	**SPSL2**	This 3 bit field defines the number of sprites requiring
01	**SPSL1**	DMA slots (number of slots divided by 2). The default is
00	**SPSL0**	8 sprites represented by binary 000.

HCNT (hex)	AA productivity	AA+ productivity	AA+ 800 x 600 SVGA
00	refresh/STRxxx	refresh/STRxxx	refresh/STRxxx
01	(unused)	sprite 0	sprite 0
02	refresh/STRLONG	refresh or floppy	refresh or floppy
03	(unused)	sprite 0	sprite 0
04	refresh	refresh or floppy	refresh or floppy
05	(unused)	sprite 1	sprite 1 (opt)
06	refresh	refresh or floppy	audio channel
07	(unused)	sprite 1	sprite 1 (opt)
08	floppy	audio channel	audio channel
09	(unused)	sprite 2	(unused)
0A	floppy	audio channel	(unused)
0B	(unused)	sprite 2	bitplane 8
0C	floppy	(unused)	bitplane 4
0D	(unused)	sprite 3	bitplane 6
0E	audio channel 0	(unused)	bitplane 2
0F	(unused)	sprite 3	bitplane 7
10	audio channel 1	(unused)	bitplane 3
11	(unused)	sprite 4	bitplane 5
12	audio channel 2	(unused)	bitplane 1
13	(unused)	sprite 4	bitplane 8
14	audio channel 3	(unused)	bitplane 4
15	(unused)	sprite 5	bitplane 6

16	sprite 0	(unused)	bitplane 2
17	(unused)	sprite 5	bitplane 7
18	sprite 0	(unused)	bitplane 3
19	bitplane 8	bitplane 8	bitplane 5
1A	bitplane 4	bitplane 4	bitplane 1
1B	bitplane 6	bitplane 6	bitplane 8
1C	bitplane 2	bitplane 2	bitplane 4
1D	bitplane 7	bitplane 7	bitplane 6
--			
4A			bitplane 1
--			(wrap)
71	bitplane 1 (wrap)	bitplane (wrap)	

COPWAITH XXX W A High order vertical bits for copper wait instruction

Bit#	Name	Definition
09	**VP9**	vertical position bit 9
08	**VP8**	vertical position bit 8
-		
01	**VE9**	vertical enable bit 9
00	**VE8**	vertical enable bit 8

BPLCON5 XXX W B Bitplane control register (extra scroll bits)

Bit#	Name	Definition
12	**PF2H8**	Playfield 2 horizontal scroll code MSB
--		
08	PF1H8	Playfield 1 horizontal scroll code MSB

4. New AA+ Display Modes

Super 72 (800 by 600 non-interlaced, 72 Hz frame rate)		bandwidth
1 or 2 Bitplanes, same as ECS, but no color fudging		2 times
3 Bitplanes	8 colors	4 times
4 Bitplanes	16 colors	4 times
5 Bitplanes	32 colors	8 times
6 Bitplanes	64 colors	8 times
7 Bitplanes	128 colors	8 times
8 Bitplanes	256 colors	8 times
6 Bitplanes EHB	32 * 2 colors	8 times
6 Bitplanes HAM	4096 colors	8 times
8 Bitplanes HAM	any of 2^24 colors	8 times
Dual Playfield, Max 4 Bitplanes per playfield		2 or 4 times

16 colors per playfield. The bank of 16 colors in the 256 color palette is selectable per playfield.

Note:

1X - requires at least normal CAS and 16 bit bus
2X - requires at least normal CAS and 32 bit bus or double CAS 16 bit bus
4X - requires at least double CAS and 32 bit bus
* - may be reduced to 2^18 = 262144 colors depending on cost of silicon
** - may be reduced to 16 or 32 bits long depending on cost of silicon

5. AA+ Chip Pin Descriptions

LEGEND:
ana = miscellaneous analog pin
anai = analog input pin
anao = analog output pin
inout = input/output pin
input = digital input pin
outpt = digital output pin
ocout = open collector output pin
power = power or ground pin

ARIEL Chip Signal Description

Signal Name	Type	Description
A_AGAIN_N	input	amiga cycle retry - inhibits increment/store
A_BLS_N	input	amiga blitter slow down - external priority
A_CS_N	input	amiga chip select
A_DATA_S_N	outpt	amiga cycle data strobe - mirrors cas during chip accesses
A_DMAL	outpt	amiga serial DMA request link - debug only
A_DONE_N	outpt	amiga cycle done - replaces DBR
A_RS	input	amiga ram versus register select
A_SYNC_N	outpt	amiga cycle sync/strobe - synchronizes ARIEL/BELLE/DEBI to nominal CCK cycle, also serves as RGA strobe
AUD_LEFT	ana o	left audio channel (analog)
AUD_RIGHT	ana o	right audio channel (analog)
B_LATCH_N	outpt	external data buffer latch enable - mirrors case during processor accesses, but inverted on read/write
B_READ_EN_N	outpt	external data buffer read (chip -> cpu) enable
B_WRITE_EN_N	outpt	external data buffer write (chip <- cpu) enable
CLK_57M	input	system clock from BELLE chip (57.27 Mhz)
DMA_ACK_N	outpt	dma cycle acknowledge
DMA_MOD_N	input	dma request modifier (end, read/write, etc...)
DMA_REQ_N	input	dma cycle request
DR_32BIT_N	input	dram is 32 bits wide (versus. 16)
DR_A_X	outpt	dram address LSB (word select)
DR_A[8:0)	outpt	multiplexed chip DRAM address bus
DR_CAS_N(0:3)	outpt	dram CAS strobe - 1 per byte (for byte write)
DR_D[31:0)	inout	chip data bus
DR_RAS(0:3)	outpt	chip DRAM row selects
DR_TR_OE_N	outpt	dram transfer (VRAM) / output enable
DR_WE_N	outpt	dram write enable
FD_RD_N	input	floppy disk read data
FD_WD_N	ocout	floppy disk write data
FD_WE_N	ocout	floppy disk write gate
INT_2_N	input	external low priority interrupt request
INT_3_N	outpt	agnus interrupt request (debug)
INT_6_N	input	external high priority interrupt request
M_CLK	outpt	mouse shift register clock (3.58 MHz)
M_DATA	input	mouse serial data input
M_LOAD_N	outpt	mouse shift register load (every 16 M_CLK)
P_16BIT_N	input	processor interface select (68000 versus. 68020 / 030
P_A(23:1)	inout	processor address bus
P_AS_N	input	processor address strobe
P_D(15:0)	inout	processor data bus
P_IPL_N(0:2)	outpt	processor interrupt priority bits
P_LDS_N	input	processor lower data strobe

P_QUAD_N	input	processor quad-word access request
P_R_W	input	processor read/write
P_UDS_N	input	processor upper data strobe
POT(0:3)	ana i	analog joystick/potentiometer inputs
SP_RxD_N	input	serial data in
SP_TxD_N	outpt	serial data out
SYS_RESET_N	input	system reset input
V_CSYNC_N	outpt	composite sync output
V_HSYNC_N	inout	horizontal sync output / genlock in
V_VSYNC_N	inout	vertical sync output / genlock in
V_LPEN_N	input	light pen strobe
VDD(0:9)	power	
VSS(0:9)	power	
AGND	power	
AVAA	power	

BELLE Chip Signal Description

Signal Name	Type	Description
A_DATA_S_N	input	chip data bus strobe - mirrors CAS during chip accesses
A_SYNC_N	input	amiga cycle sync/strobe - synchronizes ARIEL/BELLE/DEBI to nominal CCK cycle, also serves as RGA strobe
AVR,AVG,AVB	anao	analog RGB outputs
CLK_57M	outpt	system clock output (57.27 MHz)
CLK_EXT	input	external clock input (genlock 28.6 MHz)
CLK_EXT_EN_N	input	external clock enable (genlock)
CLK_OSC	input	system clock oscillator input (57.27 MHz)
DR_A_X	input	dram address LSB (word select)
DR_D[0:31)	inout	dram data bus (multiplexed with RGA)
COMP(0:1)	ana	DAC compensation loop
DAC_SCALE	ana	DAC scale factor
DAC_TEST_N	input	DAC test port enable
DAC_VREF	ana	DAC voltage reference
DV_PIXEL_CLK	outpt	digital pixel clock
DV_PIXEL_SW	outpt	pixel switch (ZD, transparency)
DVR,DVG,DVB,DVI	outpt	digital RBGI outputs
LOOP_FILTER	ana	external clock doubler loop filter
SYS_RESET_N	input	system reset input
V_CC_4	outpt	color burst / 4 (PAL composite reference / 5)
V_CCK	outpt	color clock (NTSC composite reference)
V_CSYNC_N	input	composite sync
V_CSYNC_B_N	outpt	buffered composite sync
V_CB_GATE_N	outpt	color burst gate
VSS(0:6)	power	
VDD(0:6)	power	
AGND(0:3)	power	
AVAA(0:3)	power	

DEBI Chip Signal Description

Signal Name	Type	Description
A_BLS_N	outpt	amiga blitter slowdown
A_CS_N	outpt	amiga chip select
A_DONE_N	input	amiga cycle done (was DBR)
A_SYNC_N	input	amiga clock sync / RGA strobe
BUS_OVR_N	input	expansion bus address decode override
BUS_XRDY	input	expansion busy wait
CC_A(25:22,0)	outpt	memory card address bus (high order & byte)
CC_ACK_N	input	memory card DMA acknowledge
CC_BUF_EN_N	outpt	memory card data/address buffer enable
CC_BUSY_N	input	memory card busy status/interrupt request
CC_BVD_1_N	input	memory card battery voltage/status change
CC_BVD_2_N	input	memory card battery voltage/digital audio
CC_CD_N(1:2)	input	memory card detect
CC_CEL_N	outpt	memory card chip enable low
CC_CEU_N	outpt	memory card chip enable high
CC_IORD_N	outpt	memory card IO read strobe
CC_IOWR_N	outpt	memory card IO write strobe
CC_NOISE	outpt	memory card buffered digital audio
CC_REG_EN_N	outpt	I/O versus memory card select
CC_REG_N	outpt	memory card register space select
CC_OE_N	outpt	memory card output enable
CC_REQ_N	outpt	memory card DMA request
CC_RESET	outpt	memory card reset
CC_WAIT_N	input	memory card wait request
CC_WE_PGM_N	outpt	memory card write/program enable
CC_WP_N	input	memory care write protect status
CLK_57M	input	system clock (57.2 Mhz) from BELLE
DMA_ACK_N	input	amiga DMA cycle acknowledge
DMA_MOD_N	outpt	amiga DMA cycle modifier (end, R/W, etc.)
DMA_REQ_N	outpt	amiga DMA cycle request
FD_CHNG_N	input	floppy disk change status
FD_DIR_N	outpt	floppy disk step direction select
FD_INDEX_N	input	floppy disk index pulse
FD_MTR_0_N	outpt	floppy disk (internal) motor on
FD_MTR_N	outpt	floppy disk motor control
FD_RDY_N	input	floppy disk ready status
FD_SEL_N(0:3)	outpt	floppy disk selects
FD_SIDE_N	outpt	floppy disk side select
FD_STEP_N	outpt	floppy disk step command
FD_TRK0_N	input	floppy disk track zero sense
FD_WPROT_N	input	floppy disk write protect status
FIRE_0_N	inout	joystick 0 fire button
FIRE_1_N	inout	joystick 1 fire button
FLASH_CEL_N	outpt	flash memory chip upper byte chip enable
FLASH_CEU_N	outpt	flash memory chip lower byte chip enable
IDE_CS_1_N	outpt	IDE drive chip select #1
IDE_CS_2_N	outpt	IDE drive chip select #2
IDE_IRQ	input	IDE drive interrupt request
INT_2_N	outpt	low priority interrupt request
INT_6_N	outpt	high priority interrupt request
IO_OE_N	outpt	I/O output enable
IO_READ_N	outpt	I/O read strobe
IO_RESET_N	outpt	I/O buffered reset
IO_WAIT_N	input	I/O wait request
IO_WE_N	outpt	I/O write enable

IO_WRITE_N	outpt	I/O write strobe
KB_CLOCK_N	outpt	keyboard serial clock line
KB_DATA_N	input	keyboard serial data line
KB_MPU_CLK	outpt	keyboard processor clock
KB_RESET_N	input	keyboard reset line
NET_CS_N	outpt	network controller chip select
P_A[25:8)	input	processor address bus (high order only)
P_AS_N	input	processor address strobe
P_BEER_N	trout	processor bus error
P_BG_N	input	processor bus grant
P_BGACK_N	trout	processor bus grant acknowledge
P_BR_N	outpt	processor bus request
P_CLOCK	outpt	processor clock (14 Mhz)
P_D[15:0)	inout	processor data bus
P_DTACK_N	trout	processor data transfer acknowledge
P_HLT_N	trout	processor halt
P_LDS_N	input	processor lower data strobe
P_R_W	input	processor read/write
P_RST_N	trout	processor reset
P_UDS_N	input	processor upper data strobe
PP_ACK_N	inout	parallel port acknowledge
PP_BUSY	inout	parallel port busy
PP_D[7:0)	inout	parallel port data
PP_POUT	inout	parallel port paper out status
PP_SEL	inout	parallel port select status
PP_STROBE_N	inout	parallel port strobe
PV_5	outpt	programming voltage +5 enable
PV_125	outpt	programming voltage +12 enable
PWR_LED_N	outpt	power on led
ROM_EN_N	outpt	system ROM chip enable
RTC_CS_N	outpt	real-time clock chip select
SP_CD_N	input	serial carrier-detect
SP_CTS_N	input	serial clear-to-send
SP_DSR_N	input	serial data-set-ready
SP_DTR_N	outpt	serial data-terminal-ready
SP_RTS_N	outpt	serial request-to-send
SPARE_CS_N	outpt	spare (serial) chip select
SPARE_PA_0_N	inout	spare port bit (was OVL)
SYS_RESET_N	outpt	system reset
V_HSYNC_N	input	video horizontal sync
V_VSYNC_N	input	video horizontal sync
VDD(0:7)	power	
VSS(0:7)	power	

6. AA+ Subsystem Signal Characteristics

Source/Destination Legend:

? - who knows
A - ARIEL chip
B - BELLE chip
D - DEBI chip

CC - credit card slot (PCMCIA)
DR - chip DRAM
FD - floppy disk cable / port
GP - game port
ID - IDE port
IO - I/O ?????
JU - PCB jumper
KB - keyboard connector
MO - mouse
PR - processor
SP - serial port
TS - test pin
VI - video slot / port
XP - expansion slot

Input Pins, driven external to AA+ Subsystem

Signal Name	Number	Freq Max	Source	Destination	Type	Notes
A_AGAIN_N	1	3.580	?	A	TTL	1
A_RS	1	1.750	PR	A	TTL	
BUS_OVR_N	1	1.750	XP	D	TTL	
BUS_XRDY	1	1.750	XP	D	TTL	
CC_ACK_N	1	1.750	CC	D	TTL	
CC_BUSY_N	1	1.750	CC	D	TTL	
CC_BVD_N(1:2)	2	1.750	CC	D	TTL	
CC_CD_N(1:2)	2	1.750	CC	D	TTL	
CC_WAIT_N	1	1.750	CC	D	TTL	
CC_WP_N	1	1.750	CC	D	TTL	
CLK_EXT	1	28.600	VI	B	TTL	
CLK_EXT_EN_N	1	0.000	VI	B	TTL	
CLK_OSC	1	57.200	CC	D	TTL	
DAC_TEST_N	1	0.000	TS	B	TTL	
DR_32BIT_N	1	0.000	JU	A	TTL	
DR_D(31:0)	32	14.300	DR,A,B	DR,A,B	TTL	
FD_CHNG_N	1	0.000	FD	D	TTL	
FD_INDEX_N	1	0.000	FD	D	TTL	
FD_RD_N	1	0.500	FD	D	TTL	
FD_RDY_N	1	0.000	FD	D	TTL	
FD_TRK0_N	1	0.000	FD	D	TTL	
FD_WPROT_N	1	0.000	FD	D	TTL	
IDE_IRQ	1	0.000	ID	D	TTL	
IO_WAIT_N	1	1.750	IO	D	TTL	
KB_DATA_N	1	0.010	KB	D	TTL	
KB_RESET_N	1	0.010	KB	D	TTL	
M_DATA	1	3.580	MO	A	TTL	
P_16BIT_N	1	0.000	JU	A	CMOS	
P_A[25:8)	18	1.750	PR	A,D	TTL	
P_AS_N	1	1.750	PR	A,D	TTL	
P_BG_N	1	1.750	PR	D	TTL	

P_D(15:0)	16	14.30	PR	A	TTL	
P_LDS_N	1	1.750	PR	A,D	TTL	
P_QUAD_N	1	0.000	?	A	TTL	1
P_R_W	1	1.750	PR	A,D	TTL	
P_UDS_N	1	1.750	PR	A,D	TTL	
SP_CD_N	1	0.000	SP	A	TTL	
SP_CTS_N	1	0.000	SP	A	TTL	
SP_DSR_N	1	0.000	SP	A	TTL	
SP_RxD_N	1	0.250	SP	A	TTL	
V_LPEN_N	1	0.000	GP	A	TTL	

Input Pins,driven internal to AA+ Subsystem

Signal Name	Number	Freq Max	Source	Destination	TYPE	Notes
A_S_N	1	14.300	A	B	CMOS	
A_DONE_N	1	3.580	A	D	CMOS	
A_SYNC_N	1	3.580	A	B,D	CMOS	
CLK_57M	1	57.200	B	A,D	CMOS	1
DMA_ACK_N	1	3.580	A	D	CMOS	1
DMA_MOD_N	1	3.580	D	A	CMOS	1
DMA_REQ_N	1	3.580	D	A	CMOS	
DR_A_X	1	3.580	A	B	CMOS	
SYS_RESET_N	1	0.000	D	A,B	CMOS	
V_CSYNC_N	1	0.000	A	B	CMOS	

External Analog Pins (external connection)

Signal Name	Number	Freq Max (Mhz)
AUD_LEFT	1	0.000
AUD_RIGHT	1	0.000
AVR,AVG,AVB	3	28.600
POT(0:3)	4	0.000

Internal Analog Pins (no external connection)

Signal Name	Number	Freq Max (Mhz)
COMP(1:2)	2	0.000
DAC_REF	1	0.000
DAC_SCALE	1	0.000
LOOP_FILTER	1	0.000

Output Pins

Signal Name	Number	Freq Max (Mhz)	Cap (pFd)	rise/fall (nsec)	Current DC	Current Total	Notes
A_BLS_N	1	1.750	75	5.00	0.001	0.002	
A_DMA_MOD_N	1	3.580	75	5.00	0.001	0.001	
A_DMA_REQ_N	1	3.580	75	5.00	0.001	0.001	
A_DMAL	1	3.580	75	10.00	0.001	0.001	
A_DONE_N	1	3.580	75	10.00	0.001	0.004	
A_SYNC_N	1	3.580	60	5.00	0.001	0.003	
B_LATCH_N	1	14.300	60	5.00	0.001	0.010	
B_READ_EN_N	1	1.750	60	5.00	0.001	0.002	
B_WRITE_EN_N	1	1.750	60	5.00	0.001	0.002	

CC_A[25:22,0)	5	1.750	60	5.00	0.005	0.008	
CC_BUF_EN_N	1	1.750	60	5.00	0.001	0.002	
CC_CEL_N	1	1.750	60	5.00	0.001	0.002	
CC_CEU_N	1	1.750	60	5.00	0.001	0.002	
CC_IORD_N	1	1.750	60	5.00	0.001	0.002	
CC_IOWR_N	1	1.750	60	5.00	0.001	0.002	
CC_NOISE	1	0.001	60	10.00	0.001	0.001	
CC_OE_N	1	1.750	60	5.00	0.001	0.002	
CC_REG_EN_N	1	1.750	60	5.00	0.001	0.002	
CC_REG_N	1	1.750	60	5.00	0.001	0.002	
CC_REQ_N	1	1.750	60	5.00	0.001	0.001	
CC_RESET	1	0.000	60	5.00	0.001	0.001	
CC_WE_PGM_N	1	1.750	60	5.00	0.000	0.001	
CLK_57M	1	57.200	60	5.00	0.001	0.036	
CS_NET_N	1	1.750	60	5.00	0.001	0.002	
CS_RTC_N	1	1.750	60	5.00	0.001	0.002	
CS_SPARE_N	1	1.750	60	5.00	0.001	0.002	
DMA_ACK_N	1	1.750	75	5.00	0.001	0.001	1
DR_A[0:8)	9	14.300	60	5.00	0.000	0.077	
DR_A_X	1	3.580	60	5.00	0.000	0.002	
DR_CAS(0:3)	4	14.300	76	5.00	0.000	0.043	
DR_D[0:31)	32	3.580	102	10.00	0.000	0.029	
DR_DS_N	1	14.300	97	5.00	0.000	0.014	
DR_RAS[0:3)	4	7.140	60	5.00	0.000	0.017	
DR_TR_OE_N	1	3.580	100	10.00	0.001	0.003	
DR_WE_N	1	3.580	100	10.00	0.000	0.002	
DV_PIXEL_CLK	1	28.600	60	7.50	0.001	0.018	
DV_PIXEL_SW	1	28.600	99	7.50	0.003	0.017	
DVR,DVG,DVB,DVI	4	28.600	99	7.50	0.012	0.069	
FD_DIR_N	1	0.000	102	10.00	0.012	0.012	
FD_MTR_0_N	1	0.000	60	10.00	0.003	0.003	
FD_MTR_N	1	0.000	102	10.00	0.006	0.006	
FD_SEL_N(0:3)	1	0.000	102	10.00	0.006	0.006	
FD_SIDE_N	1	0.000	102	10.00	0.006	0.006	
FD_STEP_N	1	0.000	102	10.00	0.006	0.006	
FD_WD_N	1	0.500	95	10.00	0.008	0.008	
FD_WE_N	1	0.000	95	10.00	0.008	0.008	
FIRE_N(0:1)	2	0.000	115	15.15	0.001	0.001	
FLASH_CEL_N	1	1.750	60	5.00	0.001	0.002	
FLASH_CEU_N	1	1.750	60	5.00	0.001	0.002	
IDE_CS_N(1:2)	2	1.750	60	5.00	0.002	0.002	
INT_2_N	1	0.000	60	5.00	0.002	0.002	
INT_6_N	1	0.000	60	5.00	0.002	0.002	
INT_3_N	1	0.000	60	10.00	0.000	0.000	
IO_OE_N	1	1.750	60	5.00	0.001	0.002	
IO_READ_N	1	1.750	60	5.00	0.004	0.005	
IO_RESET_N	1	0.000	125	10.00	0.003	0.003	
IO_WE_N	1	1.750	60	5.00	0.001	0.002	
IO_WRITE_N	1	1.750	60	5.00	0.004	0.005	
KB_CLOCK_N	1	0.010	60	10.00	0.001	0.001	
KB_MPU_CLK	1	3.580	60	5.00	0.001	0.003	
M_CLK	1	3.580	60	5.00	0.001	0.003	
M_LOAD_N	1	0.224	75	5.00	0.001	0.001	
P_A[1:23)	23	1.750	60	10.00	0.000	0.002	
P_BEER_N	1	1.750	60	5.00	0.000	0.000	
P_BGACK_N	1	1.750	60	5.00	0.001	0.001	
P_BR_N	1	1.750	75	5.00	0.001	0.002	
P_CLOCK	1	7.140	60	5.00	0.002	0.006	

P_D[15:0)	16	1.750	60	10.00	0.016	0.033	
P_DTACK_N	1	1.750	60	5.00	0.000	0.001	
P_HLT_N	1	1.750	60	5.00	0.000	0.001	
P_IPL_N[2:0)	3	0.000	60	10.00	0.000	0.000	
P_RST_N	1	1.750	60	5.00	0.000	0.001	
PP_ACK_N	1	0.000	100	15.15	0.003	0.003	
PP_BUSY	1	0.000	100	15.15	0.003	0.003	
PP_D[7:0)	8	0.000	101	15.15	0.096	0.096	
PP_POUT	1	0.000	101	15.15	0.003	0.003	
PP_SEL	1	0.000	101	10.00	0.003	0.003	
PP_STROBE_N	1	0.000	100	15.15	0.010	0.010	
PV_5	1	0.000	60	15.15	0.001	0.001	
PV_125	1	0.000	60	15.15	0.001	0.001	
PWR_LED_N	1	0.000	60	15.00	0.003	0.003	
ROM_EN_N	1	1.750	60	5.00	0.001	0.002	
SPARE_PA_0_N	1	0.000	60	15.00	0.003	0.003	1
SP_DTR_N	1	0.000	60	10.00	0.001	0.001	
SP_RTS_N	1	0.000	60	15.15	0.001	0.001	
SP_TxD_N	1	0.250	60	1.00	0.000	0.000	
SYS_RESET_N	1	0.000	125	10.00	0.003	0.003	
V_CB_GATE_N	1	0.056	60	7.50	0.001	0.001	
V_CC_4	1	0.850	60	10.00	0.001	0.002	
V_CCK	1	3.540	60	10.00	0.001	0.003	
V_CSYNC_B_N	1	0.056	99	25.00	0.003	0.003	
V_CSYNC_N	1	0.056	60	5.00	0.001	0.001	
V_HSYNC_N	1	0.056	95	15.15	0.005	0.005	
V_VSYNC_N	1	0.000	95	15.15	0.005	0.005	

Notes:

1 These pins are not required for the currently defined feature set and can be considered for sacrifice if a need should arise for additional pins.

7. AA System Electrical Specifications

Environmental Requirements

Characteristic	Min	Typ	Max	Units	Notes
Power Supply Voltage (VDD-VSS)	4.75	5.00	5.25	Volts	
Case Ambient Temperature	-	-	70	deg. C	
Clock Frequency (CLK_OSC/CLK_EXT)	51.5	57.2	62.9	MHz	1

Notes:

1 CLK_EXT frequency is same as CLK_OSC (57.2MHz) unless we manage to get an on-board frequency doubler working on BELLE chip, in which case frequency is 28.6MHz

DC Specifications

Characteristic	Symbol	Min	Max	Units	Notes
Digital CMOS Input High Voltage	vih01	3.67	VDD	volts	1
Digital CMOS Input Low Voltage	vil01	0	1.50	volts	
Digital TTL Input High Voltage	vih02	2.00	VDD	volts	
Digital TTL Input Low Voltage	vil02	VSS	0.80	volts	
Digital CMOS Output High Voltage	voh	VDD-0.5	-	volts	
Digital CMOS Output Low Voltage	vol	-	1.00	volts	

Notes:

1 Applies to pins: A_BLS_N, A_CS_N, A_DATA_S_N, A_RS, DMA_ACK_N, DMA_MOD_N, DMA_REQ_N

AC Specifications

#	Characteristic	Symbol	Reference	Min	Typ	Max	Units	Notes
Generic	Digital Input Setup	$ts_{,,}$	CLOCK rise	-	-	-	nSEC	
Generic	Digital Input Hold	$ts_{,,}$	CLOCK rise	-	-	-	nSEC	
Generic	Digital Input Rise/Fall	$ts_{,,}$	CLOCK rise	-	-	-	nSEC	
Generic	Digital Output Delay	$ts_{,,}$	CLOCK rise	-	-	-	nSEC	